

Comparing TUIs and GUIs for Primary School Programming

Abrar Almjally^{1,2}, Kate Howland², Judith Good²

¹ Information Technology
Al Imam Mohammad Ibn Saud Islamic University
Riyadh, Saudi Arabia
aamjally@imamu.edu.sa

² Department of Informatics
University of Sussex
Brighton, UK
{k.l.howland}{j.good}{a.almjally}@sussex.ac.uk

ABSTRACT

There is considerable interest in using tangible user interfaces (TUIs) to support teaching children programming, but evidence for the benefits is mixed, and their deployment in school environments presents more challenges than graphical user interfaces (GUIs). This study investigates the effect of GUIs and TUIs on learning outcomes, attitudes toward computing, and reported enjoyment in a computer-programming activity with primary-school students aged 6-7 in Saudi Arabia. Forty-two students engaged in a 45-minute learning activity using either a TUI or GUI programming environment. The study used a between-groups design, and quantitative data were collected, including pre-test and post-test results, and ratings on attitudinal and enjoyment surveys. Learning gains were significantly higher for the GUI group than the TUI group. However, post-activity increases in reported attitude toward computing were significantly higher for the TUI group. There was no difference in activity enjoyment scores, which were high for both groups.

CCS CONCEPTS

- Human-centered computing ~Empirical studies in HCI
- Applied computing ~Interactive learning environments
- Social and professional topics~K-12 education

KEYWORDS

Tangible interaction; Programming; Education; Robotics; Interface design.

ACM Reference format:

Abrar Almjally, Kate Howland and Judith Good. 2020. Comparing TUIs and GUIs for Primary School Programming. In *Proceedings of ACM Technical Symposium on Computer Science Education (SIGCSE'20)*, March 11--14, 2020, Portland, OR, USA. ACM,

New York, NY, USA, 2 pages.
<https://doi.org/10.1145/3328778.3366851>

1 Introduction

Teaching children programming is not a new idea; it was first championed in the 1960s and introduced to schools in the 1980s before it disappeared from primary schools for over a decade [1]. In the last ten years, computing concepts including programming have again begun to be more widely taught, and several countries in Europe, Asia, and the Middle East have introduced computer science (CS) into their school curricula, with some (including the England) now teaching programming from age five [9,18,26]. Many different approaches have been developed to support children's computing education, including CS Unplugged (<https://csunplugged.org/en/>) and graphical programming environments such as Scratch [19]. One common approach with younger children is the use of kinaesthetic activities, including tangible manipulation and physical movement, with many teachers believing that physicality can have a beneficial effect[26]. Psychologists such as Healy [8] and human-computer interaction researchers including Antle [2], have provided evidence that tangible manipulation can simplify cognitive computation, enhance understanding, and improve memory recall. Previous research has found positive results for TUIs in problem-solving [2], and reading comprehension [6], in line with theories of embodied cognition, which argue that the human body plays a significant role in thinking and acting in the world [4]. However, there is limited research examining the cognitive advantages of TUIs over GUIs for learning abstract concepts [2] such as those found in computing education (e.g. iteration and variables).

Here, we focus on TUI and GUI programming environments that use blocks-based programming. In this context, a TUI has physical blocks that users can manipulate by hand, while a GUI has virtual blocks that can be manipulated on a screen. Despite their popularity, TUIs can make it more difficult to update commands and programming structures. They can also be costly and have low portability rates [25].

This study compares the use of a TUI and GUI in learning programming skills for primary-school children aged 6-7 in Saudi Arabia. It forms part of a larger project investigating how different types of user interfaces can support the development of primary students' computing skills, and addresses the following research questions: how does interface type (TUI or GUI) affect children's

learning outcomes, attitudes towards computing, and enjoyment, in the context of a structured primary education classroom? Additionally, what are the differential gender effects in the above measures?

2 Related Work

There are some existing studies comparing TUIs and GUIs for learning programming, which have examined differences in reported attractiveness, collaboration support, programming achievement, enjoyment, usability, and gender differentials, with mixed results [23].

Horn et al. [10] conducted a between-subjects study using similar GUI and TUI interfaces in a non-school setting with 152 adults and 104 children. They found that the TUI was more inviting, supportive of collaboration, child-focused, and appealing to children under 16 years old.

Sapounidis and Demetriadis [22] explored children's preference, enjoyment, and ease with using a TUI or GUI to program a robot. All children found the TUI more enjoyable. The TUI was easier to use for younger children, especially when introducing programming concepts, although older children found the GUI easier to use. Sapounidis and Demetriadis also conducted a study in a school measuring three variables related to children's performance on tasks and four variables associated with performance during free interaction with TUI and GUI interfaces [24]. They found that the TUI users had lower error rates and engaged in more effective debugging, and younger children took less time to complete a simple sequence-programming task. In free interaction, analysis revealed that the older children in the TUI group used a wider variety of commands and parameters to extensively explore the capabilities, whilst their counterparts in the GUI group did not.

However, none of the above studies examined learning outcomes. In a study teaching programming concepts to 35 children aged 5-6, Strawhacker and Bers [27] investigated learning outcomes according to interface type: GUI, TUI, or HUI (hybrid user interface), where users can switch between the TUI and GUI interfaces. They did not find significant differences across tasks when all groups were considered, which may be due to the small sample size.

Results on gender preferences have been inconclusive. Some studies found that TUIs attract girls and boys equally [10], while GUIs appeal more to boys [22]. Another study found that girls consider TUIs more attractive [17,22]. Therefore, more investigation is needed on the influence of gender on children's interface preferences and their ability to learn computing concepts. Additionally, to allow teachers and schools to make informed decisions about the use of TUIs, more evidence is needed on the benefits for learning, as well as enjoyment and effect on attitude to computing as a subject. Thus, the current study, building on prior work [27], investigates the impact of interface type on learning, enjoyment and on attitudinal outcomes for children aged 6-7. Moreover, the research explores the relationships between these

outcomes and gender, and is the first study to compare TUIs and GUIs for learning programming in the Middle East.

3 Method

3.1 Design

The study used a between-groups design. Interface type (TUI or GUI) and gender were the independent variables, leading to four groups: TUI girls (TUI-F), TUI boys (TUI-M), GUI girls (GUI-F), and GUI boys (GUI-M). Participants completed a pre-test and attitudinal survey; a 45-minute learning activity, which included programming a robot; a post-test; and attitudinal and enjoyment surveys. The dependent variables were 1) learning gain, measured by the difference between pre-and post-test scores, 2) attitudinal change, measured by the difference in pre- and post-activity attitudinal survey scores, and 3) enjoyment, measured by participants' scores on the post-activity enjoyment survey. Participants' mean maths and science scores were added as a participant variable and used to implement matched random assignment to groups as described in the next section.

3.2 Participants and setting

The study was conducted in a Saudi Arabian primary-school during normal hours. A pre-test and pre-survey were completed in participants' classrooms. The learning activities, post-test, post-survey, and enjoyment survey were completed in a quiet room in the school. Two classes – both single gender – took part, with a total of 44 students (22 females) aged 6-7. Of these students, 42 students' parents gave informed consent for their children to participate in the study. The 42 participants were paired with same-gender students based on their average scores in mathematics and science (selected because they are related to computational thinking skills [29]). There are no standardised tests for students of this age in Saudi Arabia, so students' average coursework scores in these subjects were used. Participants were ranked based on their mean scores and assigned to conditions using blocked randomisation. Given the young age of the participants, where they asked to be paired with friends, the groups were manually adjusted while maintaining roughly equivalent scores. Table 1 shows participant distribution across conditions, including the mathematics and science mean score (M&S M) and standard deviation (M&S D).

	GUI-F	GUI-M	TUI-F	TUI-M
N	12	12	10	8
Pairs	6	6	5	4
M&S M	81.8%	84.6%	81.6%	83.1 %
M&S SD	6.28	7.21	6.61	6.30

Table 1: Group characteristics

3.3 Materials

For the learning activity we used an activity sheet, a Samsung Galaxy Tab E tablet, a robot and mat, programming tiles, and two tangible interface components: a tablet holder and a mirror that fits over the tablet's camera (similar to the Osmo system: <https://www.playosmo.com>). For data collection we also used pre and post attainment tests and attitudinal surveys, a post-activity enjoyment survey, and a GoPro video camera.

3.3.1 The system

The system is built on an existing environment, Tica [28]. It consists of TUI programming blocks, GUI blocks on the tablet's touchscreen, a physical robot that can be programmed using either language, a mat on which the robot moves, and various tiles (e.g., 'start', 'finish') that can be placed on a grid on the robot's mat to create tasks for it to complete. Both sets of blocks were designed to be as similar as possible and were the same size and colour (see Figure 1). Three action blocks are used to control the robot's movement: 1) forward (one square), 2) right turn, 3) left turn; one block for generating a sound (in this case, a buzzer); and an iteration block (which repeats the blocks placed within it two times). The TUI blocks were created with a 3D printer.



Figure 1: The TUI setting (left) and the GUI (right)

The app provides the GUI interface and is used to communicate with the robot. The app also records task data, including number of activities completed per participant, activity completion time, number of attempts made per activity, and the commands used in each attempt. The GUI has a construction pane at the top of the screen, and a block inventory at the bottom. Blocks are selected by dragging them to the construction pane. In the TUI, users select blocks by physically placing them in front of the system. The play button on each interface runs the code, and a dialogue appears to inform the user that the robot is running. Once the robot stops, the system moves to the 'task complete' screen if the goal was met or alerts the user if it was not. The robot used is an mBot, a commercial educational robot based on Arduino [15], selected for its suitability for the participants' age group. An orange arrow was added on the top of the robot and used on the activity sheet to signify the direction it faces (Figure 3). The robot moves on the mat, which has a 5x5 square grid, and five types of re-configurable task-specific tiles: 1) 'S' (the starting tile), 2) a finish flag (the end

goal), 3) a speaker (tile where the robot must stop and make a noise), 4) 'X' (a square that is blocked to the robot) and 5) grey tiles used to make a specific path for the robot to follow (see Figure 2 & Figure 3).

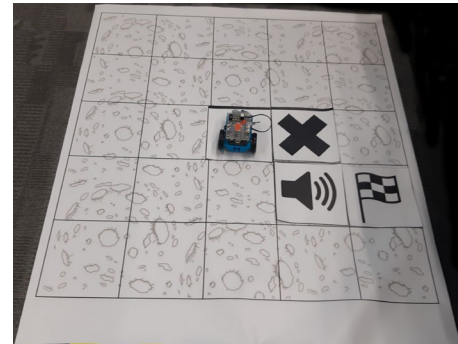


Figure 2: The mBot robot and the robot mat

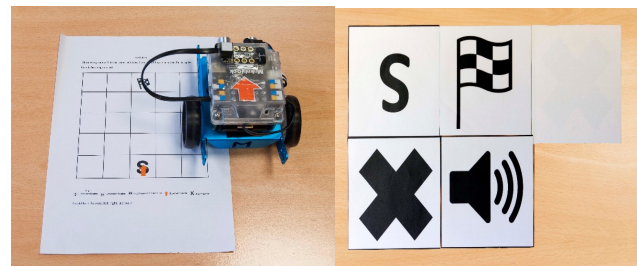


Figure 3: mBot with orange arrow and activity sheet (left); tile types (right)

3.3.2 Learning outcomes

As there is no standard primary computing curriculum in Saudi Arabia [18], the learning outcomes were based on the English National Curriculum for Computing (specifically, the Key Stage 1 curriculum), since England is one of the few countries that teaches computing in primary school as standard. The following learning outcomes were defined, targeting program creation, debugging, and comprehension.

- **Create and debug** simple programs:
 - **Describe:** articulate the program goals.
 - **Simple coding:** write a simple program by selecting the correct commands (one or more actions/types of action) and placing them correctly.
 - **Complex coding:** write an advanced program using the correct sequence of commands in one iteration.
 - **Debug:** predict what will happen, find out exactly what does happen, pinpoint and correct bugs [3].
- **Comprehension:** explain the behaviour of simple programs by identifying suitable solutions.

3.3.3 Attainment test

Two semantically identical versions of the attainment test were developed, which differed only in terms of structure of the maps. Both focused on testing the learning outcomes. Approximately half

the participants received version A as the pre-test and version B as the post-test, while the other half received the tests in reverse order, as an added measure to ensure that there were no differences between the pre and post-tests in terms of difficulty. All test images (i.e. the grids and blocks) were visually similar to the learning activity. Each test consisted of six questions (two multiple-choice questions with four potential answers and four open-ended questions) and was worth 27 points in total.

The questions (used in both versions) were as follows:

- Look at the picture below (Figure 4, left) the arrow shows where the robot is, and which way it is facing: [3 points]
 - What shape will the robot do to get to the Finish square? [Intended to familiarise participants with questions and grid]
 - How many moves to the Right will the robot have to do to go to the Finish square? [**Describe**]
 - What are the first 2 moves the robot will have to do to go to the Finish square? [**Simple coding**]
- Which of the programs below will get the robot to make noise and go to the Finish square? [1 point / **Debug**]
- Which square of the grid will the robot go to, when following the program below? Use the given sticker to show which will be the Finish square. [5 points / **Debug**]
- Write a program to make the robot go to the Finish square, use stickers [8 points / **Simple coding**]
- Now make the robot go to the Finish square using 5 blocks only, use stickers [8 points / **Complex coding**]
- Which solution is more advanced/ better and why? [1 point / **Comprehension**]

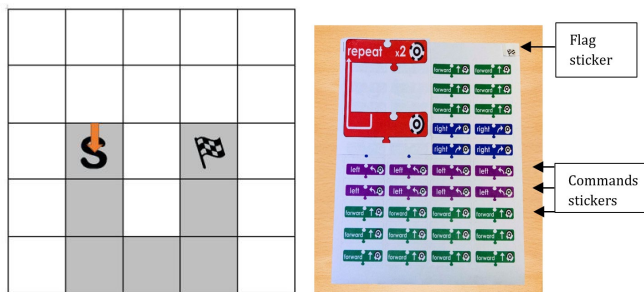


Figure 4: Example attainment test Question 1 (left) and the stickers (right)

3.3.4 Attitudinal survey

The attitudinal survey used the following five statements: 1) 'I like computing', 2) 'I am good at computing', 3) 'I like the challenge of computing', 4) 'Computing is fun', and 5) 'I want to find out more about computing'. These statements were selected – based on their relevance – from a survey used to assess the effect of a game-making project on attitudes to computing [21]. Responses were recorded on a 5-point Likert scale, presented as a series of faces ranging from sad to happy.

3.3.5 Enjoyment survey

The enjoyment survey was the same as that used in [30] and was composed of three questions. Question 1 asked how much participants liked the tool and asked them to respond using a 5-point Likert scale ('I didn't like it at all', 'I didn't like it', 'It was okay', 'I liked it', and 'I liked it very much'). Question 2 asked if participants wanted to try the activity again, and Question 3 asked if they would tell their friend about the tool: these questions had three options ('No', 'Maybe', and 'Yes'). The scales used emojis to symbolise each emotion to help the children better understand the choices on the scale.

3.3.6 Learning Activities

There were six activities with different difficulty levels. Each activity introduced or developed computing concepts, such as sequences, iteration, comprehension or debugging, and aimed to fulfil the learning outcomes. Activities were presented to participants on the activity sheet and with the robot mat using the tiles (shown in Figure 2 and Figure 3). In Activity 1, the robot must go from start to finish by moving forward four times. The aim was to familiarise students with the task and activity sheet and for them to gain confidence by solving simple problems requiring the repeated use of one type of block. In Activity 2, students must use the turn and noise commands to complete the task. In Activity 3, students must use turns and rotations to find the shortest path to the flag and avoid the 'X' tiles (e.g. Figure 2). In Activity 4, the robot must make a noise twice using the repeat command. Activity 5 requires the use of turns and the repeat command. In Activity 6, the robot has to follow the grey squares to reach the flag, requiring a more complex program that included turn blocks, some of which needed to be placed within a repeat command.

3.4 Procedure

The study ran from November to December of 2018. Participants completed two sessions. In the first session, lasting 30 minutes, students first received an introduction, and then the pre-survey was distributed, completed, and collected. The pre-test, and stickers required for answering Questions 3–5, were then distributed to participants (with one class receiving version A, and one class receiving version B). The researcher read each question and multiple-choice options aloud to the whole class, as students were not fluent in reading, but students completed their test sheets individually at their desks. No time limit was given for the tests. The researcher made sure that everyone had finished each question before moving on to the next question.

The second session took place in a quiet room in the school and lasted approximately 45 minutes. In this session, pairs of participants completed the learning activities, followed by a post-test, a post-attitudinal survey, and an enjoyment survey, which they completed individually. The procedure for both interfaces was the same: pairs of participants were first given an overview of the session and the programming environment, an explanation of the functionality of the action blocks, were asked to solve Activities 1–3, had an intervention to explain the functionality of the repeat block, and then were asked to solve Activities 4–6. However,

before starting programming, participants were required to explain orally what they wanted the robot to do, discuss their solution in pairs, and work together to program their solution. In cases of disagreement, the researcher asked one participant to explain why they thought that the chosen block was correct. Then the researcher asked the other participant if they were convinced by their partner's explanation. If so, participants were asked to choose the robot's next move and justify their choice, as before. If not, the researcher asked them to explain why they were not convinced and to propose a different solution. There were only two cases where such an intervention was required.

Participants were assisted when they needed help, with the researcher explaining the current issue and encouraging participants to find the solution themselves. The number of attempts participants could make was not limited. The researcher clapped and gave encouraging feedback each time an activity was completed, and participants generally showed enjoyment and excitement about their accomplishments. For Activity 4, involving iteration, participants first completed it without using the repeat block. Then, the researcher introduced the repeat block and explained why iteration can be useful. She asked students to solve Activity 4 again using the repeat block. For the two other activities involving iteration (Activities 5 and 6), participants were asked to solve the activity twice: once without the repeat and once with the repeat. Following the learning activities, the enjoyment survey was given, followed by the post-attitudinal survey. Then, the post-test was completed individually, with questions read aloud by the researcher, as in the procedure for Session 1.

4 Analysis

To analyse students' answers for the open-ended questions (3 to 6) we developed coding schemes for each question. A rubric was built based on a task-specific analytical rubric developed by [11,27] and errors were coded following the error analysis method used in [7]. First, each question received a total score based on placement of blocks, selection of action blocks, and level of complexity. Each correct placement of an action block scored one point, while an incorrect block received zero and was coded using the error analysis scheme. Furthermore, the repeat block was worth additional points because of its value as an advanced programming concept, and was scored according to the number of action blocks repeated. For example, in Test A, repeat iterates three action blocks, giving up to three points. Normalisation was applied to ensure that Q4 and Q5 in the two test versions were each worth eight points.

Next, the error coding scheme described in Good et al. [7] was used to look at the incorrect variants of program blocks. The rubric distinguishes between errors of omission (when blocks that should be present are left out) and errors of commission (due to the presence of blocks that should not appear because they are erroneous). Regarding the direction of the code, we read the code from top to bottom, as was demonstrated in the learning session. Finally, Question 6 asked participants to evaluate their two previous solutions in Questions 4 and 5 by selecting the better solution and giving a logical reason. All the study data were first

coded by the first author. A random sample of 20% of responses to questions 4 and 5 was generated for second coding. Inter-rater reliability was determined using Cohen's Kappa. The Kappa value was .73, indicating good agreement based on Fleiss [3].

5 Results

Due to the small sample size ($n=42$) and the nature of children's different developmental levels, some of the data did not meet all independent t-test assumptions and failed to meet normality assumptions. Therefore, we assessed the data using the Fisher skewness coefficient and Fisher Kurtosis coefficient (z values). Most of the data have an acceptable range of normality, with z values in the range of ± 1.96 , which for a small sample size ($n < 50$) is considered not significantly different from the normal distribution [12,20]. In these cases, we used parametric tests. Summary charts showing results for the tests and surveys by condition can be seen in Figure 5.

5.1 Effect of interface on learning outcomes

To measure learning outcomes, we calculated the normalised learning gains by considering the maximum possible gain or loss given the pre-test score [16], using the following formula: $100 \times (\text{post-pre}) / (100 - \text{pre})$. Before analysing the results, one outlier was identified and dealt with by adjusting the value to be closer to the rest of the data set whilst maintaining rank order, in line with the Winsorization approach [5,14]. An independent-samples t-test was run to determine differences in normalised learning gains between interface types. The GUI users had greater normalised learning gains ($M=.095$, $SD=.047$) than the TUI users ($M=.065$, $SD=.045$), a statistically significant difference of 0.293 (95% CI, .0002 to 0.058), $t(40) = 2.036$, $p=.048$.

The average normalised learning gains for the groups by gender were (M, TUI) $M=5.34$ ($SD=2.48$), (F, TUI) $M=7.5$ ($SD=5.5$), (M, GUI) $M=9.62$ ($SD=5.16$), and (F, GUI) $M=9.4$ ($SD=4.40$). A two-way ANOVA was run to examine the effects of interface and gender. The interaction effect between interface and gender on normalised learning gains was not statistically significant, $F(1, 38) = .678$, $p = .415$, partial $\eta^2 = .018$.

5.2 Effect of interface on attitudinal change

There were two outliers on attitudinal change grouped by condition, and four outliers grouped by gender; these were adjusted using the approach described in the previous section. Analysis of the attitudinal survey responses was conducted on the sum of the five questions, creating a single attitudinal measure on a 25-point maximum and 5-point minimum scale. An independent-samples t-test was run to determine if the interface type affected children's change in attitude toward computing. The TUI group showed greater increase on scores ($M = 3.169$, $SD = 3.22$) than the GUI group ($M=.958$, $SD = 3.02$), with a statistically significant difference in mean attitudinal improvement scores between the two interfaces ($M=2.208$, 95% CI, $[-.247, 4.16]$, $t(40) = 2.27$, $p=.028$).

The average attitudinal improvement for the groups by gender was (M, TUI) $M = 3.25$ ($SD = 3.89$), (F, TUI) $M = 3.1$ ($SD = 2.8$), (M, GUI) $M = .08$ ($SD = 2.88$), and (F, GUI) $M = 1.83$ ($SD = 3.04$). A two-way ANOVA was conducted to examine the effects of interface and gender. The interaction effect between interface and gender on attitudinal improvement scores was not statistically significant, $F(1, 38) = .949$, $p = .336$, partial $\eta^2 = .024$.

5.3 Effect of interface on enjoyment

As the data were not normally distributed by interface type, or by gender, a Mann-Whitney U test was used to determine if there was a difference in enjoyment scores across conditions. The GUI (mean rank = 21.9) and TUI (mean rank = 20.8) were not significantly different, $U = 204$, $z = -327$, $p = .744$.

An additional Mann-Whitney U test was run to examine differences in enjoyment scores between genders. The mean rank was not significantly different (female mean rank = 23.3, male mean rank = 19.45), $U = 179$, $z = -10154$, $p = .249$.

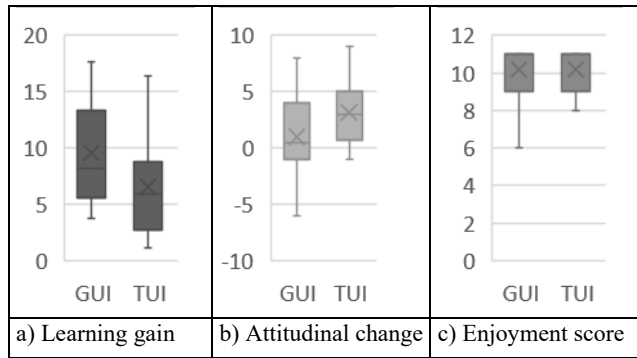


Figure 5: Summary of results by condition

6 Discussion

We compared an isomorphic TUI and GUI in a computer-programming activity with primary-school students aged 6-7 in Saudi Arabia. The study involved three dependent variables: learning gain, attitude change, and enjoyment, which were examined as a function of two independent variables (interface type and gender). There is promising existing research on the relevance of embodied cognition in other subject areas, with studies showing that physical manipulation can benefit learning in English [6] and science [13]. However, there is not yet evidence of this in CS, and our results show that GUI users had significantly greater learning gains than TUI users. This finding is also contrary to the previous study of the effects of interface type on learning outcomes [27], which found no significant difference between the groups.

On attitude towards computing measures, the TUI group had a significantly greater improvement. To understand why GUIs might support greater learning gains and TUIs might lead to attitudinal improvements, further work should carry out fine-grained examinations of interactions with both types of interface during learning activities. Despite the difference in attitude change, no

significant difference was found between interface type and enjoyment of the activities. This may be because both interfaces have the same engaging output, namely, a physical robot. Further research should investigate the role of physicality in the output as well as the programming environment. This could be examined using isomorphic interfaces with graphical programming/ graphical robot compared to tangible interaction/ physical robot.

There were no statistically significant differences between genders and interfaces for all three dependent variables, in line with [10]. There are some limitations to the study described in this paper. First, the evaluation focused on specific English Key Stage1 learning outcomes. Further work is needed to determine the extent to which each type of user interface is suited to learning other concepts in other countries' curricula. In addition, measurement of the children's learning focused on the immediate effect only, not the extent to which children could apply the concepts they had learned in a corresponding situation. Therefore, more investigation is needed to determine whether far transfer is possible and to examine how children apply knowledge to solve real computer problems rather than a paper test. A further limitation is that a 40 minute learning session is a short intervention, and a longitudinal study would be needed to determine whether the effects are the same across a longer intervention. Finally, a sample size of 42 children is appropriate for this context but does not allow for conclusive statements about the effectiveness of the TUI versus GUI. Further work should target a larger sample size.

7 Conclusion

Our findings suggest that GUIs can be more effective in teaching young children fundamental programming concepts, while TUIs can have a greater positive effect on children's attitudes towards computing. The results illustrate how different interfaces could be of benefit depending on the primary goal of educators. For example, if the aim is to improve attitudes towards computing, this might justify the additional cost and effort required to deploy TUIs in schools. This work is the first step in examining the benefits of different interface types to enhance teaching programming concepts to young children in Saudi Arabia. In the next stage, we will review the video data gathered to analyse the differences in how participants use the interfaces in an attempt to understand why the use of the TUI leads to a greater improvement in attitudes towards computing.

8 Ethics

Parents gave written consent for all participants to take part in the study, after reading an information sheet. Participants additionally gave verbal assent in response to an assent script prior to taking part in research activities

ACKNOWLEDGMENTS

The authors wish to thank the school administration, teachers, and children who gave their time and effort for our study.

REFERENCES

- [1] P. Albion. 2016. *The second coming of coding: will it bring rapture or rejection?* Australian Council for Computers in Education 2016 Conference (ACCE 2016), Brisbane, Australia.
- [2] Alissa N. Antle. 2013. Exploring how children use their hands to think: An embodied interactional analysis. *Behav. Inf. Technol.* 32, 9 (September 2013), 938–954. DOI:https://doi.org/10.1080/0144929X.2011.630415
- [3] Joseph L. Fleiss, Bruce Levin, and Myunghee Cho Paik. 2004. The Measurement of Interrater Agreement. In *Statistical Methods for Rates and Proportions* (Third). John Wiley & Sons, Inc, 598–626. DOI:https://doi.org/10.1002/0471445428.ch18
- [4] Jennifer M.B. Fugate, Sheila L. Macrine, and Christina Cipriano. 2018. The role of embodied cognition for transforming learning. *International Journal of School and Educational Psychology*, 1–15. DOI:https://doi.org/10.1080/21683603.2018.1443856
- [5] Dhiren Ghosh and Andrew Vogt. 2012. Outliers: An Evaluation of Methodologies.
- [6] Arthur M Glenberg, Andrew B Goldberg, and Xiaojin Zhu. 2010. Improving early reading comprehension using embodied CAI. *Instr. Sci.* 39, 1 (2010), 27–39. DOI:https://doi.org/10.1007/s11251-009-9096-7
- [7] Judith Good, Katy Howland, and Keiron Nicholson. 2010. Young people's descriptions of computational rules in role-playing games: An empirical study. In *Proceedings - 2010 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2010*, 67–74. DOI:https://doi.org/10.1109/VLHCC.2010.18
- [8] J.M. Healy. 1998. Failure to connect: How computers affect our children's minds—and what we can do about it. *New York: Touchstone*, 350.
- [9] Fredrik Heintz, Linda Mannila, and Tommy Farnqvist. 2016. A review of models for introducing computational thinking, computer science and computing in K-12 education. *Proc. - Front. Educ. Conf. FIE 2016-Novem*, (October 2016), 1–9. DOI:https://doi.org/10.1109/FIE.2016.7757410
- [10] Michael S. Horn, Erin Treacy Solovey, R. Jordan Crouser, and Robert J.K. Jacob. 2009. Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 975. DOI:https://doi.org/10.1145/1518701.1518851
- [11] Anders Jonsson and Gunilla Svingby. 2007. The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review* 2, 130–144. DOI:https://doi.org/10.1016/j.edurev.2007.05.002
- [12] Hae-Young Kim. 2013. Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis. *Restor. Dent. Endod.* 38, 1 (February 2013), 52–4. DOI:https://doi.org/10.5395/rde.2013.38.1.52
- [13] Carly Kontra, Daniel J. Lyons, Susan M. Fischer, and Sian L. Beilock. 2015. Physical Experience Enhances Science Learning. *Psychol. Sci.* 26, 6 (June 2015), 737–749. DOI:https://doi.org/10.1177/0956797615569355
- [14] Sang Kyu Kwak and Jong Hae Kim. 2017. Statistical data preparation: management of missing values and outliers. *Korean J. Anesthesiol.* 70, 4 (2017), 407. DOI:https://doi.org/10.4097/KJAE.2017.70.4.407
- [15] Make block. 2018. mBot | Makeblock. *Makeblock*. Retrieved February 25, 2019 from https://www.makeblock.com/steam-kits/mbot
- [16] Jeffrey D. Marx and Karen Cummings. 2007. Normalized change. *Am. J. Phys.* 75, 1 (January 2007), 87–91. DOI:https://doi.org/10.1119/1.2372468
- [17] M. McLean and D. Harlow. 2017. Designing inclusive STEM activities: A comparison of playful interactive experiences across gender. In *Proceedings of the 2017 ACM Conference on Interaction Design and Children*. DOI:https://doi.org/10.1145/3078072.3084326
- [18] Ministry of Education; Tatweer Company for Educational Services. 2018. iEN National Education Portal. courses. Retrieved August 15, 2019 from https://ienbooks.t4edu.com/#/courses/58
- [19] MIT. 2003. Scratch. *MIT*. Retrieved January 31, 2018 from https://wiki.scratch.mit.edu/wiki/Development_of_Scratch_1.0
- [20] Marjorie A. Pett. 2016. Evaluating the Characteristics of Data. In *Nonparametric Statistics for Health Care Research*. 17–46.
- [21] Judy Robertson. 2013. The influence of a game making project on male and female learners' attitudes to computing The influence of a game making project on male and female learners' attitudes to computing. *Comput. Sci. Educ.* 13, March (2013), 1–41. DOI:https://doi.org/10.1080/08993408.2013.774155
- [22] Theodosios Sapounidis and Stavros Demetriadis. 2013. Tangible versus graphical user interfaces for robot programming: exploring cross-age children's preferences. *Pers. Ubiquitous Comput.* 17, 8 (December 2013), 1775–1786. DOI:https://doi.org/10.1007/s00779-013-0641-7
- [23] Theodosios Sapounidis and Stavros Demetriadis. 2017. Educational robots driven by tangible programming languages: A review on the field. *Advances in Intelligent Systems and Computing* 560, 205–214. DOI:https://doi.org/10.1007/978-3-319-55553-9_16
- [24] Theodosios Sapounidis, Stavros Demetriadis, and Ioannis Stamelos. 2015. Evaluating children performance with graphical and tangible robot programming tools. *Pers. Ubiquitous Comput.* 19, 1 (January 2015), 225–237. DOI:https://doi.org/10.1007/s00779-014-0774-3
- [25] Theodosios Sapounidis, Ioannis Stamelos, and Stavros Demetriadis. 2016. Tangible User Interfaces for Programming and Education: A New Field for Innovation and Entrepreneurship. 271–295. DOI:https://doi.org/10.1108/S2051-229520160000002016
- [26] Sue Sentance and Andrew Csizmadia. 2017. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Educ. Inf. Technol.* 22, 2 (March 2017), 469–495. DOI:https://doi.org/10.1007/s10639-016-9482-0
- [27] Amanda Strawhacker and Marina U Bers. 2015. "I want my robot to look for food": Comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. *Int. J. Technol. Des. Educ.* 25, 3 (2015), 293–319. DOI:https://doi.org/10.1007/s10798-014-9287-7
- [28] John Wilkie and Judith Good. 2017. Tica: An environment for exploring tangible vs. screen-based programming | Psychology of Programming Interest Group. In *PPIG 2017 - 28th Annual Conference*, 9.
- [29] Jeannette M Wing. 2010. Computational Thinking: What and Why? (2010).
- [30] Nesra Yannier. 2016. *Bridging Physical and Virtual Learning: A Mixed-Reality System for Early Science*.